Elsevier Editorial System(tm) for Image and Vision Computing
Manuscript Draft

Manuscript Number: IMAVIS-D-10-00271R2

Title: A pointwise smooth surface stereo reconstruction algorithm without correspondences

Article Type: Full Length Article

Keywords: Stereo vision; photogrammetry; 3D reconstruction; medical applications

Corresponding Author: Dr Richard Brown, Ph.D.

Corresponding Author's Institution: University of Canterbury

First Author: Richard Brown, Ph.D.

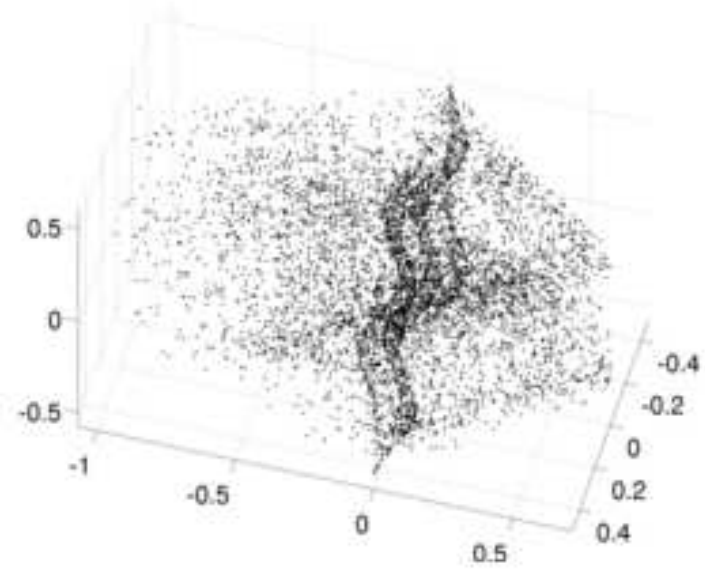
Order of Authors: Richard Brown, Ph.D.; J. Geoffrey Chase; Chris E Hann

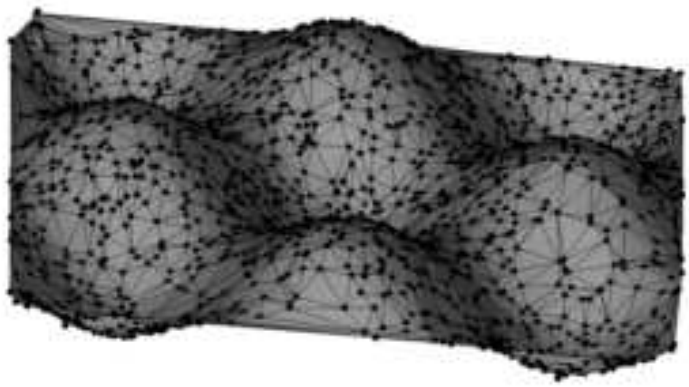
Abstract: This paper describes an algorithm for 3D reconstruction of a smooth surface with a relatively dense set of self-similar point features from two calibrated views. We bypass the usual correspondence problem by triangulating a point in space from all pairs of features satisfying the epipolar constraint. The surface is then extracted from the resulting point cloud by taking advantage of the statistical and geometric properties of the point distribution on the surface. Results are presented for computer simulations and for a laboratory experiment on a silicon gel phantom used in a breast cancer screening project.

Abstract

This paper describes an algorithm for 3D reconstruction of a smooth surface with a relatively dense set of self-similar point features from two calibrated views. We bypass the usual correspondence problem by triangulating a point in space from all pairs of features satisfying the epipolar constraint. The surface is then extracted from the resulting point cloud by taking advantage of the statistical and geometric properties of the point distribution on the surface. Results are presented for computer simulations and for a laboratory experiment on a silicon gel phantom used in a breast cancer screening project.

***Graphical Abstract**





***Research Highlights**

* 3D surface reconstruction from 2D point sets without correspondences
* Surface extraction from space of all potential epipolar reconstructions*
* Application to breast cancer screening

Response to reviewers

Richard Brown

August 13, 2011

I have included the requested responses to reviewers comments below. The (unabridged) reviewer's comments are in clear text, my responses are in italics.

> Reviewer #3: The authors have addressed the majority of comments raised during the first review.
>
> Unfortunately, the validation using real experiments has not really been enhanced. Section 4.6 is quite vague and no meaningful numbers are reported. Specifically:
>
> The authors state "reasonably large number of points" .. How many?

The answer is actually not yet particularly well known, so the requirements specified were set to be relatively high (O(thousands) of points). I have tried to clarify this in the final sentence of the first paragraph of Section 4.6.

> Figure 9 should show all the original images used for the reconstruction (all 5 of them). Perhaps with the points used highlighted. Are there any outliers at all?

Figure 9c now includes the set of five images, and also shows by colour each point on each image used in the 3D reconstruction. There are outliers. I have manually checked every correspondence generated by the algorithm (by visually inspecting 100x100 pixel windows surrounding the two relevant points - the unused colour information came in useful here for manual checking!) and the results are summarised in Table 2.

> I don't see the relevance of warping the images by a homography - it is not related at all to this work - and it does not highlight the improved matching in difficult images.

Fair enough - I was trying to show visual evidence that almost all correspondences were correctly computed. Table 2 is a much better way to get my point across.

> - The authors mention that normal matching approaches wont work in this case but this is not demonstrated. Was this actually attempted? A comparison to illustrate the advantage of using the proposed technique would have been very helpful.

A short comparison illustrating how SIFT features are unsuitable for the correspondence problem for the types of images has been added at the end of §4.6. Various methods were attempted and discarded early on in the project - it is possible that with some sustained effort that customisation of some existing technique could have been made to work. However, the decision was made to develop a new method based on the geometry that would be robust to changes in the imaging conditions - i.e. the correspondence algorithm is entirely decoupled from any image processing, and use the available prior information in a more natural way.

> - The authors report that most matches are correct but some are outliers - the actual figures need to be provided. How many used? How many matched? How many outliers?

See above comment - this has been rectified

> - Was this experiment tried only on one dataset? It would have been good to try more.

Yes, and the algorithm is also now being used on human subjects. I've provided a reference, and mentioned that similar results have been obtained in other similar laboratory experiments.

> Minor point - Some of the language in the article has become quite casual, more in the style of an informal project report rather than a technical publication. In particular, project specific issues should be removed and the focus should be on the technical contribution.

I've tried to change the language in some parts of the article where I thought this was an issue - the introduction has been slightly reworked, as has some of the discussion. I hope I've managed to resolve this in a satisfactory manner.

A pointwise smooth surface stereo reconstruction algorithm without correspondences

R. G. Brown[a,1,*], J. G. Chase[b], C. E. Hann[b]

[a]*Department of Mathematics and Statistics, University of Canterbury, Private Bag 4800, Christchurch 8140, New Zealand*
[b]*Department of Mechanical Engineering, University of Canterbury, Private Bag 4800, Christchurch 8140, New Zealand*

Abstract

This paper describes an algorithm for 3D reconstruction of a smooth surface with a relatively dense set of self-similar point features from two calibrated views. We bypass the usual correspondence problem by triangulating a point in space from all pairs of features satisfying the epipolar constraint. The surface is then extracted from the resulting point cloud by taking advantage of the statistical and geometric properties of the point distribution on the surface. Results are presented for computer simulations and for a laboratory experiment on a silicon gel phantom used in a breast cancer screening project.

Keywords: Stereo vision, photogrammetry, 3D reconstruction, Medical applications

[*]Corresponding author
Email address: R.Brown@math.canterbury.ac.nz (R. G. Brown)
[1]Phone: +64 3 3642987 ext 7678, Fax: +64 3 3642587

1. Introduction

Why write another article on yet another method for 3D reconstruction from digital image pairs? This article addresses a specific type of problem that may be encountered in some applications for which standard techniques are not appropriate or are difficult to use. The problem is that of reconstructing in 3D a relatively dense set of specific points on a smooth surface from two calibrated views, but in a situation where solving the correspondence problem is difficult due to a high degree of self-similarity between the image features.

The motivation for this research came from a problem in medical imaging. In the DIET (Digital Image-based Elasto-Tomography) project [1], it is necessary to reconstruct in 3D the motion of a large set of points on a human breast surface as it is being mechanically vibrated. The motion of specific points on the surface is required, rather than a description of the surface motion as a whole, because this point motion is required to solve an inverse problem for the 3D distribution of internal elasticity of the breast tissue.

Because in this type of application, specific feature points are required that can be tracked as the surface deforms, volume-based methods [2–4] based on visual hulls that do not measure the location of trackable individual surface points are not appropriate. Likewise, state of the art stereo methods such as graph cuts [5, 6] and belief propagation [5, 7] that are based on Markov Random Fields are also inappropriate, as they are concerned with creating a 3D reconstruction of a scene by computing a pixel-by-pixel depth map, and once again do not compute the precise locations of trackable individual surface points. For the DIET problem small artificial fiducials are randomly applied to the breast surface. For practical reasons these features are essentially identical in appearance, and hence difficult to

correspond using standard interest-point detectors [8–10]. Dense stereo methods [11, 12] relying on accurate correspondence of a number of key features are therefore also not well-suited to this application. Dellaert *et. al.* [13] have developed an algorithm for computing feature point correspondences without any *a priori* information by computing the maximum likelihood estimate of the scene and cameras using an EM algorithm. This method is potentially a promising alternative approach, however significant work would be required to modify the algorithm to incorporate large numbers of occluded features and an arbitrary number of world features.

The types of application envisioned for the algorithm presented herein are those where a (relatively) dense set of self-similar points on a smooth surface needs to be reconstructed in 3D from fully calibrated cameras. Another application might be in surveying, for example, where landmarks such as trees or plants are the feature points, and the goal is to reconstruct the topography on which they lie from multiple aerial views.

2. Preliminaries

2.1. Camera model

The computer vision notation used follows that of Hartley and Zisserman [14], a standard reference for multiple view geometry. In this paper, cameras are fully calibrated, and a 3×4 camera projection matrix $\mathtt{P}$ is assumed to be of the form

$$\mathtt{P} = \mathtt{K}\begin{bmatrix}\mathtt{R} & \mathbf{t}\end{bmatrix} \tag{1}$$

where $\mathtt{R} \in \mathbb{R}^{3\times3}$ is a rotation matrix, $\mathbf{t} \in \mathbb{R}^3$ a 3-vector, and $\mathtt{K} \in \mathbb{R}^{3\times3}$ an upper triangular matrix representing the internal parameters of the camera. The projection,

in homogeneous coordinates, from world coordinates $\mathbf{X} = (X, Y, Z, W)^\mathsf{T}$ to image coordinates $\mathbf{u} = (u, v, w)^\mathsf{T}$ is therefore described by the linear equation

$$\lambda \mathbf{u} = \mathtt{P}\mathbf{X} \tag{2}$$

where λ is a nonzero scalar. The image coordinates $\mathbf{u}$ can be *wlog* be scaled so that $w = 1$, in which case u, v are the pixel coordinates.

It is often more convenient to work in normalised homogeneous coordinates

$$\mathbf{x} = \mathtt{K}^{-1}\mathbf{u} \tag{3}$$

which if scaled to the form $(x, y, 1)$ represent coordinates on the plane $Z_c = 1$ in the camera's 3D Euclidean frame of reference. For a typical camera, the relationship between $(x, y, 1)$ and $(u, v, 1)$ is given by

$$\begin{aligned} u &= \alpha_x x + u_c \\ v &= \alpha_y y + v_c \end{aligned} \tag{4}$$

where α_x, α_y are scale parameters, typically very similar in magnitude, and u_c, v_c are the image centre.

2.2. Epipolar Constraint

Let $\mathbb{I}_1, \mathbb{I}_2$ be two image frames of reference corresponding to cameras with projection matrices $\mathtt{P}_1, \mathtt{P}_2$. The camera poses induce an epipolar geometry, whereby a potential match $\mathbf{v} \in \mathbb{I}_2$ for a point $\mathbf{u} \in \mathbb{I}_1$ is constrained to lie on a particular epipolar line in $\mathbb{I}_2$. The epipolar line has homogeneous coordinates $\mathbf{l}$, such that $\mathbf{l}^\mathsf{T}\mathbf{v} = 0$ for all points $\mathbf{v}$ on the line. The epipolar line $\mathbf{l} \in \mathbb{I}_2$ corresponding to a given point $\mathbf{u} \in \mathbb{I}_1$ is given by $\mathbf{l} = \mathtt{F}\mathbf{u}$ where $\mathtt{F} \in \mathbb{R}^{3\times3}$ is the fundamental matrix, which can be computed directly from the two projection matrices $\mathtt{P}_1$ and $\mathtt{P}_2$ [14].

In the absence of noise, points $\mathbf{u} \in \mathbb{I}_1$ and $\mathbf{v} \in \mathbb{I}_2$ therefore satisfy the epipolar constraint if

$$\mathbf{u}^\mathsf{T}\mathtt{F}^\mathsf{T}\mathbf{v} = 0. \tag{5}$$

In reality, image point locations and the fundamental matrix itself are subject to noise and so (5) is in general not satisfied for corresponding points. For a noisy image point measurement $\hat{\mathbf{u}}$, the corresponding point will not necessarily lie on the line $\mathtt{F}\hat{\mathbf{u}}$, but will fall within the envelope of a small range of epipolar lines. If measurement errors are small and the epipole is located well outside the image plane, as is the case in most stereo vision applications including the camera geometry used in the DIET project, this range of lines will be approximately parallel. The epipolar constraint in the presence of noise can hence be written

$$d_\perp(\mathbf{v}, \mathtt{F}\mathbf{u}) < \delta \tag{6}$$

where $d_\perp(\mathbf{v}, \mathbf{l})$ is the perpendicular Euclidean distance between the point $\mathbf{v}$ and the line $\mathbf{l}$ in image space.

The threshold δ can easily be estimated by Monte Carlo simulation as follows. Let $\mathbf{u} \in \mathbb{I}_1$ and $\mathbf{v} \in \mathbb{I}_2$ be corresponding (noise-free) points and assume that the image measurement errors are zero-mean, isotropic, and Gaussian, with variance σ_m^2 in each coordinate. A small random error $\mathbf{w}_u$ sampled from this noise distribution is added to an image point $\mathbf{u} \in \mathbb{I}_1$, and the resulting epipolar line $\mathbf{l} = \mathtt{F}(\mathbf{u} + \mathbf{w}_u)$ is recorded. This process is repeated a number of times. The result will be a band of near-parallel lines. Define σ_l^2 to be the resulting variance in the distance of each line from the central (mean) line. Assuming that this distribution is also Gaussian, the distribution of $d_\perp(\mathbf{v} + \mathbf{w}_v, \mathtt{F}(\mathbf{u} + \mathbf{w}_u))$ will have variance $\sigma_l^2 + \sigma_m^2$, where $\mathbf{w}_v$, the error in $\mathbf{v}$, is drawn from the same noise distribution. Using three standard

deviations as the threshold, the matching threshold is hence:

$$\delta = 3\sqrt{\sigma_m^2 + \sigma_l^2}. \tag{7}$$

This threshold is not necessarily consistent for all points **u** over the entire image, so a pragmatic approach is to compute δ for a range of image points $\mathbf{u}_k$, and to choose the largest.

2.3. Triangulation

Triangulation is the process of constructing a 3D point from two image points. See the text by Hartley and Zisserman [14, Chapter 12] for a good overview. In the presence of noise, triangulation of two image points is approximately the midpoint of the shortest line segment joining the two skew rays from the optical centres through the normalised image points.

2.3.1. Error analysis

An important question is how much image measurement errors affect the accuracy of the triangulation. First, assume that the calibration matrix equation is of the form (4), with $\alpha_x = \alpha_y = \alpha$ and that image measurement errors have zero mean and variance σ_m^2 in each (pixel) coordinate. This translates to a variance of $\left(\frac{\sigma_m}{\alpha}\right)^2$ in each normalised (x, y) coordinate. Normalised coordinates will be used for the remainder of this section.

Consider a world point **X** and its two projections $\mathbf{x}_1$ and $\mathbf{x}_2$ in two images. The epipolar plane formed by the two optical centres, $\mathbf{O}_1$, $\mathbf{O}_2$, and **X** induces an epipolar line in each image. Let $\mathbf{e}_1$ and $\mathbf{e}_2$ be unit vectors directed from each optical centre to **X**, and let d_1 and d_2 be the Euclidean distances from each optical centre to **X**. Finally, let ϕ be the magnitude of the angle $\mathbf{O}_1\mathbf{X}\mathbf{O}_2$. These quantities are depicted in Fig. 1.

Let $\mathbf{x}_1$ be subject to measurement error, as above, and let $\mathbf{x}_2$ remain noise-free. It is convenient to split this error into two orthogonal components, one aligned with the epipolar line and one normal to it. Represent these components by two Gaussian random variables $W_t, W_n \sim \mathcal{N}\left(0, \left(\frac{\sigma_m}{\alpha}\right)^2\right)$.

A small deviation of W in normalised coordinates corresponds to an angular deviation $\delta\theta$ satisfying $W\cos^2\theta \leq \delta\theta \leq W\cos\theta$, where θ is the acute angle between the image ray and the normal to the image plane. For a typical camera, the values of θ are relatively small, so let $\delta\theta \approx W$. This value of $\delta\theta$ is a slight overestimate, but considerably simplifies the computations that follow. The same analysis can easily be repeated with more precise $\delta\theta$ values if required. In components,

$$\begin{aligned} \delta\theta_t &\approx W_t \\ \delta\theta_n &\approx W_n. \end{aligned} \tag{8}$$

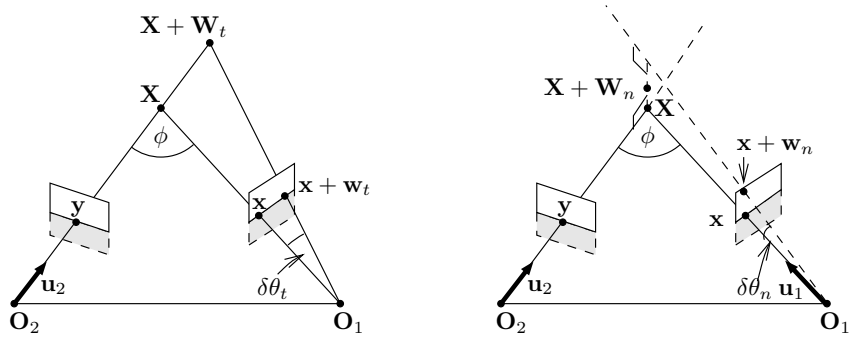
It is now possible to work out how much the deviations $\delta\theta_t, \delta\theta_n$ contribute to error in the triangulation.

Consider first the angular deviation along the epipolar line $\delta\theta_t$. With reference to Fig. 1a, it can be shown that the induced error $\mathbf{W}_t$ in the triangulation is given by

$$\mathbf{W}_t \approx \left(\frac{d_1\delta\theta_t}{\sin\phi}\right)\mathbf{u}_2, \tag{9}$$

showing that the magnitude of the error is dependent on the angle between the camera directions as well as the distance from the camera. Consider now the angular deviation normal to the epipolar line $\delta\theta_n$. With reference to Fig. 1b, it can be show that the induced error $\mathbf{W}_n$ in the triangulation is given by

$$\mathbf{W}_n \approx \left(\frac{d_1\delta\theta_n}{2}\right)\mathbf{u}_1 \times \mathbf{u}_2. \tag{10}$$



(a) Measurement error component along epipolar line

(b) Measurement error component normal to epipolar line

Figure 1: Triangulation error induced by measurement error in one image by components normal and tangential to epipolar line

as the triangulation is approximately the midpoint of the shortest line joining the skew rays. Unlike $\mathbf{W}_t$, this error is not dependent on the angle between the cameras and is relatively smaller than $\mathbf{W}_t$.

Now let both image points $\mathbf{x}_1, \mathbf{x}_2$ be subject to error. The error in the triangulation is given by

$$\begin{aligned}\mathbf{W} &= \mathbf{W}_t^{(1)} + \mathbf{W}_n^{(1)} + \mathbf{W}_t^{(2)} + \mathbf{W}_n^{(2)} \\ &= \left(\frac{d_1\delta\theta_t^{(1)}}{\sin\phi}\right)\mathbf{u}_2 + \left(\frac{d_2\delta\theta_t^{(2)}}{\sin\phi}\right)\mathbf{u}_1 \\ &\quad + \left(\frac{d_1\delta\theta_n^{(1)} + d_2\delta\theta_n^{(2)}}{2}\right)\mathbf{u}_1 \times \mathbf{u}_2 \\ &= U_1\mathbf{e}_1 + U_2\mathbf{e}_2 + U_3\mathbf{e}_3\end{aligned} \tag{11}$$

where $\mathbf{e}_1, \ldots, \mathbf{e}_3$ are the three orthonormal coordinate directions

$$\mathbf{e}_1 = \frac{\mathbf{u}_1 + \mathbf{u}_2}{\|\mathbf{u}_1 + \mathbf{u}_2\|} \qquad \mathbf{e}_2 = \frac{\mathbf{u}_1 - \mathbf{u}_2}{\|\mathbf{u}_1 - \mathbf{u}_2\|} \qquad \mathbf{e}_3 = \mathbf{u}_1 \times \mathbf{u}_2 \tag{12}$$

and where the components U_1, U_2, U_3 are zero-mean Gaussian random variables with variances σ_i^2 given by

$$\begin{aligned}
\sigma_1^2 &= \left(\frac{\sigma_m}{\alpha}\right)^2 \left(d_1^2 + d_2^2\right)\left(\frac{1+\cos\phi}{2\sin^2\phi}\right) \\
\sigma_2^2 &= \left(\frac{\sigma_m}{\alpha}\right)^2 \left(d_1^2 + d_2^2\right)\left(\frac{1-\cos\phi}{2\sin^2\phi}\right) \\
\sigma_3^2 &= \left(\frac{\sigma_m}{\alpha}\right)^2 \left(\frac{d_1^2 + d_2^2}{4}\right).
\end{aligned} \tag{13}$$

The error is greatest in the direction $\mathbf{u}_1 + \mathbf{u}_2$, for acute ϕ, and least in the direction orthogonal to $\mathbf{u}_1$ and $\mathbf{u}_2$, and the variance in direction $\mathbf{u}_1 + \mathbf{u}_2$ has a minimum when $\phi = \frac{\pi}{2}$. The error increases approximately linearly with the distance from the feature point to the cameras.

It will be useful later to know the magnitude of triangulation error in a particular direction. The component of triangulation error in the direction of a unit vector $\mathbf{n}$ is a zero-mean Gaussian with variance

$$\sigma_n^2 = \left\| \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \begin{bmatrix} \mathbf{e}_1^\mathsf{T} \\ \mathbf{e}_2^\mathsf{T} \\ \mathbf{e}_3^\mathsf{T} \end{bmatrix} \mathbf{n} \right\|^2 = \|\mathcal{B}\mathbf{n}\|^2 \tag{14}$$

3. Method

3.1. Problem Definition

Let $\mathcal{S} \subset \mathbb{R}^3$ be the surface in Euclidean space to be reconstructed and let $\mathcal{X} = \{\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N\}$ be the set of feature points on $\mathcal{S}$. The feature points are points on the surface that can be reliably extracted and localised in images, but are indistinguishable in appearance from one another. Make the following two assumptions about $\mathcal{S}$:

Assumption 1 (Smoothness)**.** *The surface $\mathcal{S}$ has continuous directional derivatives, and the maximum principal curvature over the entirety of $\mathcal{S}$ is bounded by some known constant κ*

Assumption 2 (Feature distribution)**.** *The feature points are distributed over the entire surface at some approximately known mean density ρ. Moreover they are distributed in such a way that there is little aggregation, or patchiness.*

Let two cameras be defined by their projection matrices $\mathtt{P}_1, \mathtt{P}_2$, assumed to be of the form (1). Each camera images only a subset of $\mathcal{X}$, so let $\mathcal{I}_1, \mathcal{I}_2$ be the indices of the points imaged by each camera, yielding two sets of image points

$$\begin{aligned} \mathcal{A}_1 &= \{\mathtt{P}_1\mathbf{X}_i + \text{noise} \mid i \in \mathcal{I}_1\} \\ \mathcal{A}_2 &= \{\mathtt{P}_2\mathbf{X}_i + \text{noise} \mid i \in \mathcal{I}_2\} \end{aligned} \tag{15}$$

The noise is considered to be Gaussian with variance σ_m in each coordinate, as discussed in §2.3.

The mathematical problem is to estimate as many points of $\mathcal{X}$ as possible, from image coordinates $\mathcal{A}_1$ and $\mathcal{A}_2$, the two projection matrices $\mathtt{P}_1$ and $\mathtt{P}_2$, and using assumptions 1 and 2, with no knowledge of the index sets $\mathcal{I}_1$ and $\mathcal{I}_2$.

3.2. Algorithm Overview

The problem being attempted in feature-point based 3D reconstruction is essentially the *correspondence problem*. Once point correspondences have been identified, 3D triangulation is a simple computation (see §2.3). However, finding feature correspondences for the problem under consideration is not straightforward. Only the epipolar constraint (§2.2) can be used to directly limit the number of feature correspondences as features are otherwise indistinguishable.

When identifying potentially corresponding points to an image point $\mathbf{u} \in \mathcal{A}_1$, the epipolar constraint finds points that lie in a narrow band about the epipolar line computed from the point. The narrower the band, the fewer points will be in it, so one way to narrow down the list of potential feature correspondences is to invest more effort in minimising model and measurement error. This is, at its limit, an impractical approach as it is subject to the limitations of the imaging equipment.

In order to progress with the problem, more information is thus required. This information comes in the form of the two assumptions, Assumptions 1 and 2, about the geometry of the surface in $\mathbb{R}^3$. A direct way to incorporate this information is to work in $\mathbb{R}^3$, rather than working directly with the two image projections.

The essential idea behind the proposed method is to construct a 3D point for each potential feature correspondence generated by the epipolar constraint. The result is a large cloud of points that contains the surface points and a large number of outliers. This cloud is referred to as the epipolar cloud. The problem then becomes that of extracting the surface points from this cloud of points. The advantage of this approach is that the two geometric assumptions can be incorporated directly in the search for surface points, freeing the problem from the projective distortion imposed by the imaging process.

The assumptions mean that surface points in a ball of suitably small radius r centred on a surface point $\mathbf{X}$, will fall approximately on a plane, while the outliers in the ball from the cloud will be randomly scattered. Therefore, a plane fitting procedure that is robust to outliers can be used to fit a plane through $\mathbf{X}$ to the ball and the inliers to the plane fit are likely also to be surface points. If $\mathbf{X}$ is not a surface point, the plane that is fit will be randomly oriented.

A suitable choice of the radius r of the balls needs to be determined. The

information about feature density can be used for this purpose. The radius r needs to be sufficiently high that the r-ball around each surface point contains enough other surface points to reliably fit the tangent plane. The mean number, n_B, of surface point neighbours within distance r of $\mathbf{X}$ needs to be high enough so that there is a very small chance of there being too few surface points in any r-ball, but exactly how high this needs to be depends on the spatial distribution of the points on the surface. A sensible choice is $n_B = 12$, see §4.4.1. If the mean point density is known to be ρ pts per unit area, the radius r is then given by

$$r = \sqrt{\frac{n_B}{\rho\pi}}. \tag{16}$$

What is still required is a means of linking points together. Planes fit through to neighbouring surface points should vary in normal direction by only a small amount, depending on the curvature κ. Therefore points are linked together if they are inliers to each others' plane fit and have normals which are sufficiently closely aligned. The surface points will then be a connected component in the resulting connectivity graph.

3.3. Epipolar cloud construction

The first stage of the algorithm is to construct the *epipolar cloud*. All pairs of points $\mathbf{x} \in \mathcal{A}_1, \mathbf{y} \in \mathcal{A}_2$ satisfying the noisy epipolar constraint (6) with threshold δ, are identified, and for each correspondence, a 3D point is constructed by triangulation, together with its associated error matrix $\mathtt{B}$, as discussed in §2.3. The matching threshold δ, is assumed to have been estimated in advance, as discussed in §2.2. The point cloud formation process is illustrated in Fig. 2. The shape and density of the resulting cloud depend on the camera locations, and on the epipolar matching threshold δ.

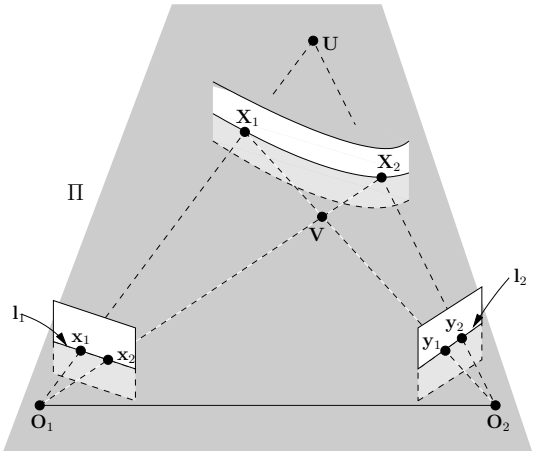


Figure 2: Epipolar constraint and point cloud formation. Any choice of two points on $\mathbf{l}_1$ and $\mathbf{l}_2$ will satisfy the epipolar constraint (6). In this image, the two world points $\mathbf{X}_1, \mathbf{X}_2$ generate two image points in each image. Each pair of points $(\mathbf{x}_1, \mathbf{y}_1)$, $(\mathbf{x}_1, \mathbf{y}_2)$, $(\mathbf{x}_2, \mathbf{y}_1)$, $(\mathbf{x}_2, \mathbf{y}_2)$ satisfy the epipolar constraint, and the resulting constructed points in space are the true surface points $\mathbf{X}_1, \mathbf{X}_2$, together with two additional points $\mathbf{U}, \mathbf{V}$ resulting from incorrect correspondences.

3.4. Plane fitting

Once the epipolar cloud has been constructed, the next stage of the algorithm is to attempt to fit tangent planes to each surface point in the cloud.

3.4.1. Finding inliers to a tangent plane estimate

Let $C = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_N\}$ denote the set of points in the epipolar cloud, and $\mathtt{B}_1, \ldots, \mathtt{B}_N$ the associated error matrices. Define $B_r(\mathbf{X}) = \{\mathbf{Y} \in \mathbb{R}^3 \mid \|\mathbf{Y} - \mathbf{X}\| < r\}$ to be the r-ball centred on $\mathbf{X}$.

Consider a surface point $\mathbf{X}_i \in \mathcal{X}$, and let T, with normal $\mathbf{n}$ be the tangent plane to $\mathcal{S}$ at $\mathbf{X}_i$. Given that the maximum principal curvature of the surface is κ, the perpendicular deviation of another nearby surface point $\mathbf{X}_j$ from T is bounded by

$$d_\kappa < \frac{\left\|\mathbf{X}_j - \mathbf{X}_i\right\|^2 \kappa}{2} \tag{17}$$

Now, let $\mathbf{Y}_k \in C$ be the corresponding point in the epipolar cloud to $\mathbf{X}_i$. The other surface point estimates (not outliers) will deviate from T according to the curvature bound above, and triangulation error. Assuming that the corresponding error matrix $\mathtt{B}_k$ is constant throughout the ball, the normal deviations from T due to triangulation error will be normally distributed with variance $\|\mathtt{B}_k\mathbf{n}\|^2$. $\mathbf{X}_i$ is, however, unknown, so any estimate of T that is computed will not pass directly through it. As will be seen in §3.4.2, the tangent plane estimates $\hat{T}_k$ with normals $\hat{\mathbf{n}}_k$ used in this algorithm are constrained to pass through $\mathbf{Y}_k$ and so the normal deviations from $\hat{T}_k$ due to noise will include the error in $\mathbf{Y}_k$ and will hence be normally distributed with variance

$$\sigma^2_{\text{noise}} = 2\,\|\mathtt{B}_k\hat{\mathbf{n}}_k\|^2 \tag{18}$$

Using $3\sigma_{\text{noise}}$ as the noise threshold, for an estimate $\hat{T}_k$ of the tangent plane to $\mathcal{S}$, passing through a point $\mathbf{Y}_k \in C$, a point $\mathbf{Y}_j$ is therefore potentially a surface point, or inlier if its perpendicular distance $d_\perp(\mathbf{Y}_j, \hat{T}_k)$ from $\hat{T}_k$ is

$$d_\perp(\mathbf{Y}_j, \hat{T}_k) < \frac{\left\|\mathbf{Y}_k - \mathbf{Y}_j\right\|^2 \kappa}{2} + 3\sqrt{2}\,\|\mathtt{B}_k \hat{\mathbf{n}}_k\| \tag{19}$$

3.4.2. Tangent plane estimation

Consider a point $\mathbf{Y}_k \in C$, and assume it corresponds to a true surface point, rather than being an outlier. Let $\mathbf{B}$ be the set of points of C within $B_r(\mathbf{Y}_k)$. A method is required for estimating the tangent plane $\hat{T}_k$ to $\mathcal{S}$ at $\mathbf{Y}_k$ from the points of $\mathcal{B}$ where there are potentially many outliers. The tangent plane estimate is constrained to pass through $\mathbf{Y}_k$.

The, perhaps, most obvious approach to consider is to fit the plane using l_1 minimisation. l_1 minimisation is reasonably robust to the presence of outliers, however it has an undesirable property when constrained to pass through a particular point. The optimal l_1 fit will pass exactly through $\mathbf{Y}_k$, and two other points of C. These two points, even if they are surface points, will produce a skewed estimate of the tangent plane. To see why, imagine estimating the tangent line to the unit circle at (1, 0) by intersecting the point (1, 0) with another point anywhere on the circle. The tangent line estimate will always be skewed in one particular direction. The same situation applies here if fitting a tangent plane using only two additional points. Ideally, the plane should be fit to three or more additional surface points by least squares, but still constrained to pass through $\mathbf{Y}_k$.

The stochastic method RANSAC (RANdom Sample And Concensus) [15] is instead used. Sets of m points from $\mathcal{B}$ are randomly sampled, and a plane fit by least squares, constrained to pass through $\mathbf{Y}_k$, to these m points. The number of

inliers satisfying (19) are counted, and after a suitably large number of samples, the plane estimate with the greatest number of inliers is refined by fitting by least squares the entire inlier set. The number of samples to take to ensure having sampled m inliers with probability p can be estimated dynamically:

$$N_{\text{iterations}} = \left\lceil \frac{\log(1-p)}{\log(1-w^m)} \right\rceil \tag{20}$$

where w is the estimated proportion of inliers, equal at any point in the algorithm to the greatest number of inliers found to that point, divided by the number of points in $\mathcal{B}$. Typically, $p = 0.99$ is used.

The best fit plane is found, if possible, using this method for every point in the cloud. The inliers to the final plane fit, together with the normal to the plane are stored for later use. Experimentation shows that there is little or no gain in choosing a value of $m > 3$.

3.5. Surface Extraction

Assuming that the tangent plane fitting procedure has worked correctly, neighbouring surface points should have tangent plane normals that are very similar in orientation. Assuming maximum principal curvature κ, the normals $\mathbf{n}_1, \mathbf{n}_2$ of the planes fit to two points separated by Euclidean distance d, will have an angular separation γ satisfying

$$\cos\gamma = |\mathbf{n}_1 \cdot \mathbf{n}_2| \geq \left(1 - \frac{d^2\kappa^2}{2}\right). \tag{21}$$

Each point in the cloud is treated as the node of a graph $\mathcal{G}$. Points $\mathbf{Y}_i, \mathbf{Y}_j$ are then connected by an (undirected) edge if and only if

1. $\mathbf{Y}_j$ is an inlier to the tangent plane estimate for $\mathbf{Y}_i$

2. $\mathbf{Y}_i$ is an inlier to the tangent plane estimate for $\mathbf{Y}_j$
3. $\mathbf{Y}_i$ and $\mathbf{Y}_j$ have normals $\mathbf{n}_i, \mathbf{n}_j$ satisfying (21)

Neighbouring surface points should then always be connected, and the surface points should therefore be one large connected component of $\mathcal{G}$. The planes fitted to the outlying points will be oriented essentially randomly, and there should hence be no other large connected components. The connected components of $\mathcal{G}$ can be found using a standard depth-first search.

The estimate of the surface point set $\mathcal{X}$ is therefore taken to be the largest connected component of the graph $\mathcal{G}$. More generally, a surface may be piecewise smooth. In this case, instead of simply choosing the largest connected component, all sufficiently large connected components are considered to comprise the surface.

3.6. Duplicate Removal

It is possible during the surface extraction phase that two or more of the extracted 3D points correspond to the same image point from one of the images, *i.e.* that the correspondences identified are not one to one. Depending on the application it may or may not be necessary to remove these. A heuristic approach that works fairly well is the following procedure that chooses the point that is best approximated by its neighbours.

1. For each point $\mathbf{u}$ in $\mathcal{A}_1$ that has multiple correspondences:
    - For each 3D point $\mathbf{Y}$ constructed from correspondences with $\mathbf{u}$, find all neighbours within distance r of $\mathbf{Y}$ that are not constructed from $\mathbf{u}$ and fit a plane to these neighbours by least squares

- The point **Y** that is closest to the corresponding plane is chosen to be the correct correspondence

2. Repeat the procedure for the second image

4. Numerical Experiments

4.1. Test surface

A number of experiments were performed using simulated data. For a test surface, the function

$$z = \alpha \cos 3\pi x \sin 3\pi y, \qquad x, y \in [-0.5, 0.5] \tag{22}$$

was used. Points were randomly generated on the surface by choosing x and y from $\mathcal{U}(-0.5, 0.5)$ and generating the corresponding z values. This gives a slight irregularity to the surface, but this effect is not large. This function has maximum principal curvature given by

$$\kappa_{\max} = 9\pi^2\alpha \tag{23}$$

4.2. Implentation Details

All code in this section was written purely in MATLAB and experiments were run on a single 1.2GHz core of an Intel Core Duo laptop with 2GB RAM. The experiments were run using a MATLAB R2010b installation in a 32 bit Linux environment. The experiments involving parameter sweeps were run inside a parfor loop provided by the Parallel Processing Toolbox, and hence the code was not utilising the built-in multithreading provided by a standard MATLAB environment. Speed was not the objective, so the code has not been particularly optimised. See §5 for discussion on how the code could be optimised for performance. Note for

each of the experiments, except the laboratory case study, the duplicate removal procedure was not carried out, so as to give a fairer indication of the performance of the surface extraction process.

4.3. Example

This example will illustrate the performance of the algorithm under typical conditions. 2000 points were randomly scattered, as discussed previously, on the test surface (22) with $\alpha = 1/\pi^2$. The resulting surface has maximum principal curvature 9, and surface area 1.20, and is depicted in Fig. 3a. Two cameras were placed, directed towards the centre of the surface, at a distance of 5 units from the surface, and with angular separation 30°. The overall geometry is depicted in Fig. 3b. Each camera was assigned the following calibration matrix

$$\mathrm{K} = \begin{bmatrix} 8000 & 0 & 800 \\ 0 & 8000 & 800 \\ 0 & 0 & 1 \end{bmatrix}$$

such that the image points fell in an approximately 1600×1600 pixel region. Random Gaussian noise with standard deviation $\sigma_m = 0.2$ pixels was applied to each coordinate. This level of error is consistent with what can easily be achieved by using blob centroids as feature location measurements. The epipolar threshold was computed by Monte-Carlo simulation to be $\delta = 0.87$ pixels. The noisy epipolar constraint (6) then generated 6328 potential correspondences, each of which was triangulated to produced the epipolar cloud depicted in Fig. 3c. Tangent planes were then fitted to the points, as described in §3.4, using the parameters $\kappa = 9$, $n_B = 12$ and $\rho = 1663$. Planes and normals were successfully fit to 3010 of the 6632 points. The connectivity graph $\mathcal{G}$ was then constructed.

The largest connected component, by number of nodes, of $\mathcal{G}$ comprised 2061 points, of which 1988 were triangulations from correct correspondences, and 73 were from false correspondences. Of the false matches, the maximum distance of the triangulated point from the true surface was 0.0227 and the mean was 0.0063. To put this into context, the height of each "hump" of the surface is approximately 0.1. Therefore almost all of the false points are indistinguishable from the true surface points. Note that the algorithm does not necessarily find all of the true surface points, the randomness of the point distribution can mean that certain regions don't have sufficient points to correctly estimate the tangent plane.

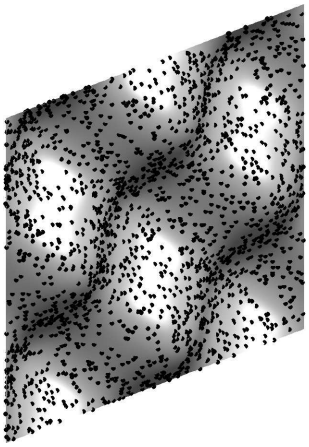
The reconstructed surface was meshed using a Delaunay triangulation of its x and y coordinates to aid visualisation, and the result is depicted in Fig. 3d

4.4. Sensitivity to Parameters

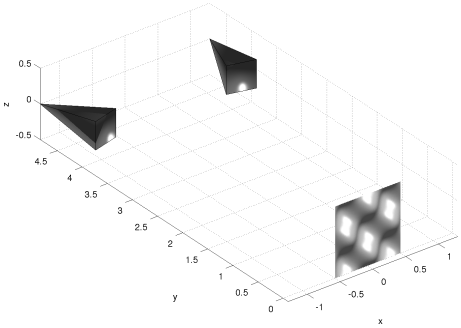
The essential parameters that govern the performance of the surface extraction part of the algorithm are the ball radius r and the maximum curvature estimate κ. The radius r is computed from a user estimate of the user estimate of surface point density, ρ, using (16), for some value of n_B. The algorithm needs to be robust to significant errors in ρ.

4.4.1. Algorithm performance under errors in ρ and choosing n_B

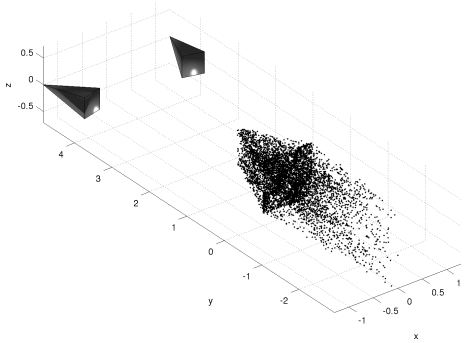
One issue that needs to be addressed is how best to work out r from an estimate ρ of surface point density, or equivalently to ascertain the best value of n_B from (16) such that the algorithm is relatively robust to errors in the estimate of ρ. An experiment was conducted using the same parameters as §4.3, but using values of n_B between 0 and 30. For each value of n_B, the algorithm was run with the true density $\rho = 1663$, and also for $\rho \pm 30\%$. Each experiment was conducted on the



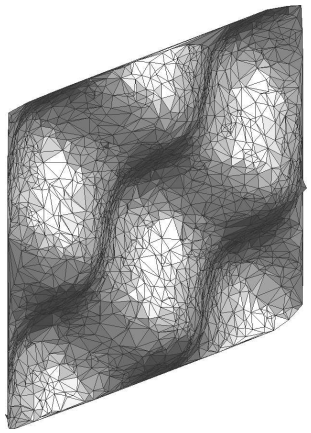
(a) Test surface, with features shown, $\kappa_{max} = 9$



(b) Experiment geometry



(c) Epipolar cloud



(d) Reconstructed surface, extracted from cloud

Figure 3: Surface reconstruction process from synthetic model

same epipolar cloud constructed from a surface with 2000 points. The following measures were used to evaluate algorithm performance:

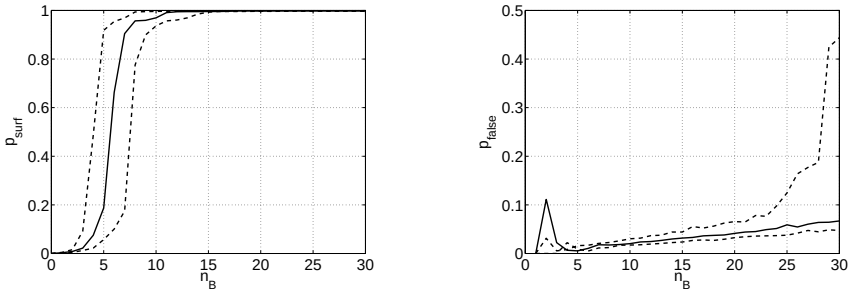
$$\begin{aligned} p_{\text{surf}} &= \frac{\#\text{ of true correspondences}}{\#\text{ of true surface points}} \in [0,1] \\ p_{\text{false}} &= \frac{\#\text{ of false correspondences}}{\#\text{ of reconstructed surface points}} \in [0,1] \\ \bar{e} &= \text{Mean dist of false points from } \mathcal{S} \end{aligned} \tag{24}$$

The quantity p_{surf} is a measure of the proportion of the surface reconstructed, the quantity p_{false} is a measure of what amount of the reconstruction is from false correspondences, and the quantity $\bar{e}$ is a measure of the reconstruction error from false correspondences. These quantities were computed for the reconstructions from each individual experiment.

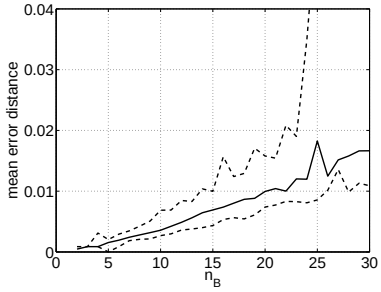
The three resulting curves are plotted in Fig. 4. These plots indicate the performance of the algorithm under errors in ρ for any given value of n_B by reading the graph vertically. Both the proportion of points from false correspondences, p_{false}, and the mean error increase with increasing n_B. The proportion of the surface reconstructed, p_{surf} drops quickly to near zero when n_B gets too small. Therefore, the choice of n_B is a tradeoff between surface reconstruction and error. From the graphs, a choice of n_B between 12 and 15 gives reasonable performance for this example under variation in ρ, and has proven suitable in other numerical experiments.

4.4.2. Algorithm performance under errors in κ

The other parameter that is user-estimated is the maximum principal curvature κ of the surface. This is difficult to reliably estimate, so it is important that the algorithm is not particularly sensitive to errors in this quantity. To test the effects of variability in κ an experiment was performed with the same parameters

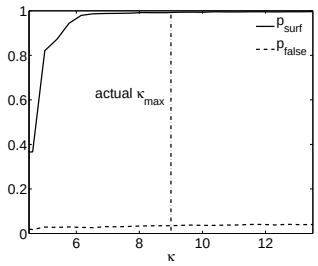


(a) p_{surf} for varying n_B (b) p_{false} for varying n_B

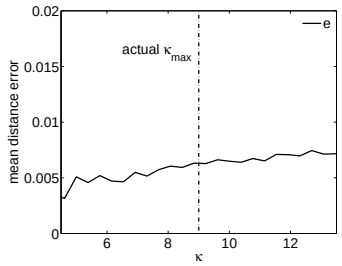


(c) $\bar{e}$ for varying n_B

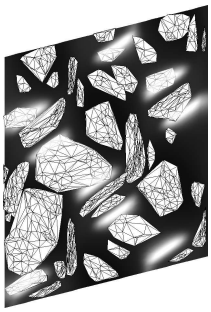
Figure 4: Performance measures for a range of r values. The three curves on each panel are for n_B computed from (16) using $\rho \pm 30\%$ (dashed) and ρ (solid) where $\rho = 1663$ is the true mean density for the surface used



(a) p_{surf}, p_{false} for varying κ



(b) $\bar{e}$ for varying κ



(c) Components of size > 10, $\kappa = 2$

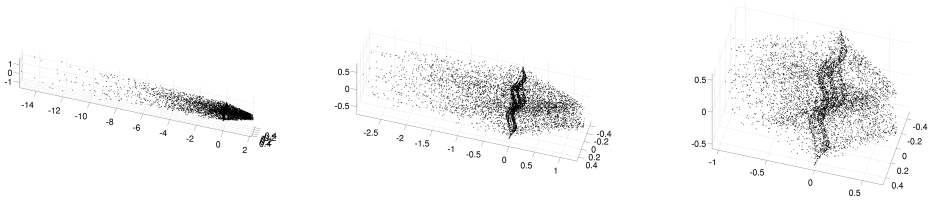
Figure 5: Performance measures for a range of κ values where the true $\kappa = 9$

as in §4.3, except that κ varied between ±50% of its true value of 9. The same error measures (24) were recorded for each parameter combination. The results are depicted in Fig. 5. Note that κ has very little effect on the performance of the algorithm for this example, until it drops below a critical value at around 6. In this case, the algorithm cannot join surface patches along a high-curvature ridge, and it splits into multiple smooth components. The mean false point reconstruction error, $\bar{e}$ increases slightly with increasing κ, as an increase in κ increases the distance threshold for plane fitting, as in (19), and hence slightly more distant outliers are included as inliers to the tangent plane fit. The algorithm therefore performs satis-

factorily over almost the entire range of $\kappa \pm 50\%$. Note that if κ is much lower than the curvature present on the surface, the algorithm will still reconstruct locally flat patches. For example, Fig. 5c depicts all the connected components of size 10 or greater constructed if κ is set to 2 for this example. Note that a reasonable portion of the surface is constructed, despite the low curvature threshold. This property of the algorithm is beneficial if the surface under consideration is piecewise smooth.

4.4.3. Algorithm performance under different camera angles

The shape, size, and density of the epipolar cloud is highly dependent on the poses of the two cameras. Cameras that have small angular separation and small baseline will tend to create a long, relatively sparse cloud, while two cameras with large baseline will create a short, more dense cloud. However, small baseline reduces the reconstruction accuracy (§2.3, so choosing the best camera positions is a tradeoff between algorithm performance and accuracy. The same experiment was repeated five times: the two cameras were placed on a circle of radius five, centred on the origin (the centre of the surface), with the angular separation of the camera varying between $\theta = 15^\circ$ and 120°. Fig 6 shows the geometry of the epipolar cloud for each value of θ. The larger the separation of the cameras the smaller and more dense the epipolar cloud. The algorithm performance for each case is detailed in Table 1. It can clearly be seen that performance drops off as the cloud becomes more dense; the proportion of the surface reconstructed drops slightly, the proportion of reconstruction from false matches increases, and the mean error from false points increases. These results suggest that in designing an experiment it is best to have the cameras relatively near to one another.



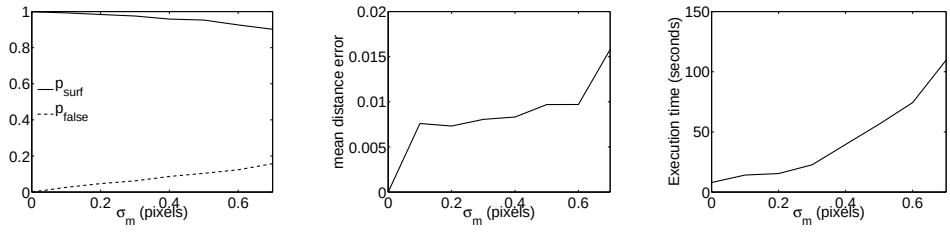
(a) $\theta = 15°$ (b) $\theta = 30°$ (c) $\theta = 60°$

(d) $\theta = 90°$ (e) $\theta = 120°$

Figure 6: Epipolar cloud geometry for varying degrees of camera separation

θ	p_{surf}	p_{false}	$\bar{e}$
15°	1.00	0.016	0.0050
30°	0.99	0.027	0.0044
60°	0.99	0.048	0.0046
90°	0.97	0.159	0.0214
120°	0.95	0.498	0.0179

Table 1: Error performance under varying camera angular separation θ



(a) $p_{\text{surf}}, p_{\text{false}}$ for varying σ_m (b) $\bar{e}$ for varying σ_m (c) run time for varying σ_m

Figure 7: Performance measures for a range of image noise values $0 \leq \sigma_m \leq 1$

4.4.4. Algorithm performance under differing image noise levels

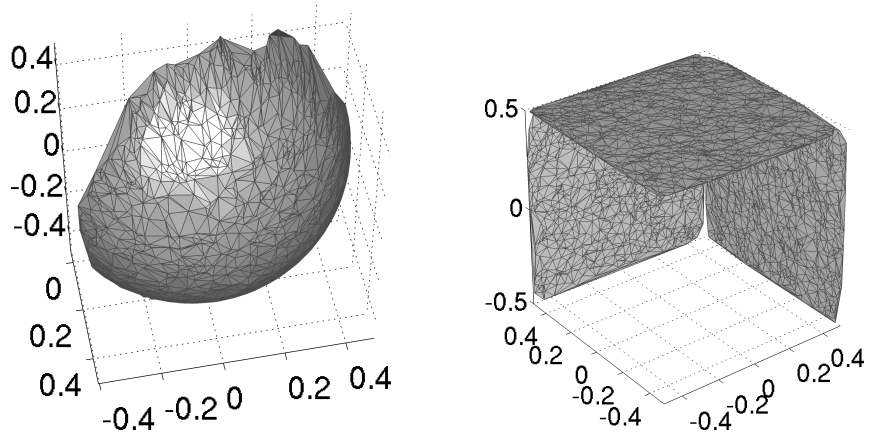
The parameter that has the most effect on algorithm performance is the noise in the image point measurements. The number of correspondences generated by the epipolar constraint is linear in σ_m, so doubling σ_m approximately doubles the density of the epipolar cloud, significantly impeding algorithm performance. Experiments were performed, using the same parameters as in §4.3, but varying the amount of image noise in the images using values $0 \leq \sigma_m \leq 1$ pixel. The results are shown in Fig. 7. For $\sigma_m = 0$, the epipolar cloud size is 2000, as would be expected, the cloud is simply the 2,000 true points. For $\sigma_m = 1$, however, the epipolar cloud has 24,000 points, whereby each point has on average 11 false correspondences. Fig. 7c shows the execution times for the range of σ_m values. The error measures for the range of σ_m values are shown in Fig. 7, where it can be seen that the various error measures increase smoothly with increasing σ_m.

4.5. Further examples

Two further examples are briefly presented here, to demonstrate the algorithm's performance in two different types of situation. The first example, de-

picted in Fig. 8a is the reconstruction of a portion of a sphere from two views. This example highlights how the algorithm performs when a reasonable proportion of the surface points are not present in both views. This surface is also of similar geometry to the breast-screening application. To allow for noise in the fitted normal directions, the maximum curvature parameter κ was set to be 25% higher than the true curvature of the sphere. The sphere had a radius of 0.5, and comprised 3142 randomly distributed points, giving a density of $\rho = 1000$ points per unit area. The remaining parameters were the same as in §4.3. 1244 points were reconstructed, of which 1165 were true surface points, giving the parameters $p_{\text{surf}} = 0.37$ (most of the points are occluded), and $p_{\text{false}} = 0.064$. The mean error $\bar{e}$, measured as distance from the true sphere, was 2.3×10^{-3} and the largest error of a reconstructed point was 7.0×10^{-3}.

The second example, depicted in Fig. 8b is the reconstruction of three faces of a cube from two views. Instead of taking the largest connected component, instead the union of all components with 20 or more points is taken to be the reconstructed surface. The diagram shows the resulting reconstruction, and the three components that were found by the algorithm. Each face of the cube was 1 unit long, and there were 1000 true points per face. The maximum curvature parameter was set to be $\kappa = 1$, to allow for a small amount of noise in the normal orientation, and all other parameters were the same as for the example in §4.3. The algorithm reconstructed 2968 points, of which 2953 were true surface points giving $p_{\text{surf}} = 0.98$, $p_{\text{false}} = 0.0051$. The errors were $\bar{e} = 1.3 \times 10^{-3}$ with maximum error 2.9×10^{-3}. Note, however, that the algorithm was not originally developed to be used in this way, and a subject of future work is to extend the algorithm to the reconstruction of piecewise smooth surfaces, paying close attention to correctly



(a) Sphere reconstruction (b) Cube reconstruction

Figure 8: Additional numerical experiments

fitting points near surface boundaries.

4.6. Case Study: DIET breast-screening system

In the DIET breast cancer screening system, the algorithm is used to reconstruct the surface of a breast in 3D from pairs of calibrated views. The feature points are defined to be the centroids of small artificially applied fiducial markers, designed to be easy to segment from the background by a simple thresholding procedure [16]. The project requires a relatively high density of feature points to be reconstructed, as it uses the motion of individual points as the input to an inverse problem that reconstructs the internal stiffness of the breast tissue [1]. The required density is not yet well known, but the algorithm needs to be able to reconstruct hundreds to thousands of points.

The method is tested in the laboratory on silicon gel breast phantoms, designed to have similar mechanical properties to human breasts. Small markers are randomly applied to the surface, these are shown in an image of the phantom in

Fig. 9a. Note that the points are coloured but that the colour information is not used at all in the results presented here. Five cameras are positioned around the phantom in such a way that any point on the phantom surface is viewed by two overlapping cameras. The camera geometry is depicted in Fig. 9b. Each camera is fitted with a custom strobing flash that allows images to be taken at any phase angle of the periodic motion. The cameras are fully calibrated, including relative position, with a custom calibration object [16].

The imaging procedure takes images of the phantom over one period of motion, at 20 equally spaced time instants. The strobe lighting ensures that the images are perfectly synchronised. The features are extracted by thresholding and then computing the centroid, which is estimated to have zero-mean Gaussian error of standard deviation $\sigma_m \approx 0.2$ pixels in each coordinate direction [16]. Tracking of the features is trivial, the motion between frames is sufficiently small relative to the point spacing that nearest neighbour matching is sufficient. Therefore all that is required for the 3D motion reconstruction is to use the algorithm in this paper to compute a 3D reconstruction, and hence identify feature correspondences, from a single time instant.

The feature density was estimated to be 10 pts/cm^2, or $\rho = 10 \times 10^4$ pts/m^2. A lower bound on radius of curvature was estimated to be 3cm, and hence $\kappa = 1/0.03 \approx 33$. The algorithm was used with image pairs from each pair of adjacent cameras to reconstruct a portion of the surface, and the final reconstruction is simply the union of the five individual reconstructions. The five images used are shown in Fig. 9c, together with the points that were used for each of the five pairwise reconstructions.

Measuring the 3D departure from some ground truth is not a useful test of the

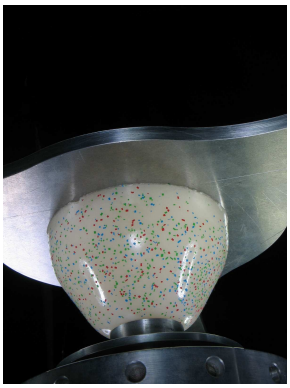
performance of the algorithm, as those errors are determined primarily by the accuracy of the imaging system itself, such as the camera calibration and the feature point localisation procedure. A better measure of the algorithm performance is to identify the proportion of correct correspondences generated by the algorithm. Each of the correspondences computed by the algorithm was manually checked by visually comparing 100×100 pixel windows around each point from the potential correspondence. The results are summarised in Table 2. One of the camera pairs, $3 \leftrightarrow 4$ performed poorly, with only a small number of correspondences being identified. This was due to one of the cameras being slightly out of calibration, which caused matching points in some regions not to satisfy the epipolar constraint. Despite this, the proportion of correct correspondences remained high at around 98%. Typically if the algorithm fails, it is by not reconstructing a portion of the surface, rather than generating false correspondences. Of all of the incorrect correspondences, only a small number are clearly visible by eye in the 3D reconstruction. This is because misidentified points still sit very near the surface.

For the purposes of visualisation, the surface point cloud was triangulated by projecting it onto a sphere and computing the Delaunay triangulation of the azimuth and altitude coordinates. The resulting triangulated surface is depicted in Fig. 9d.

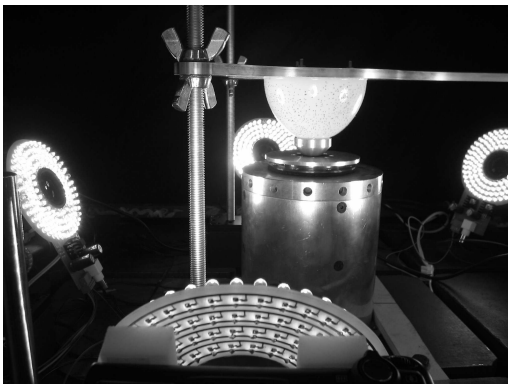
Results from other imaging experiments, including on human subjects, yield similar performance. See [17] for more details of the use of the algorithm in this breast cancer screening system.

4.6.1. Comparison with SIFT matching

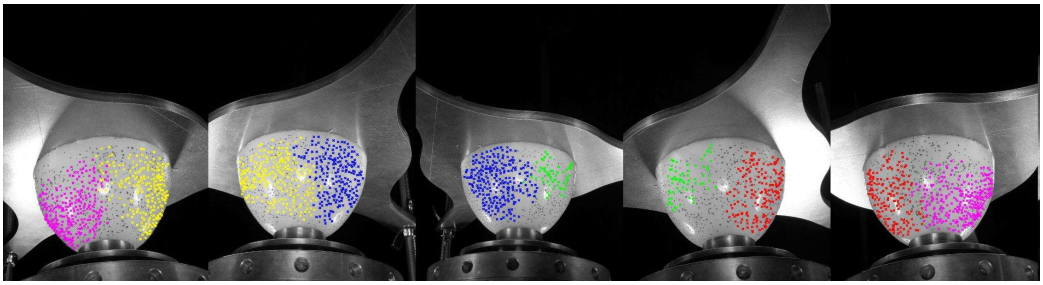
As an illustrative comparison, image features were identifed, and attempted to be matched using the Scale-Invariant Feature Transform (SIFT) [8]. The im-



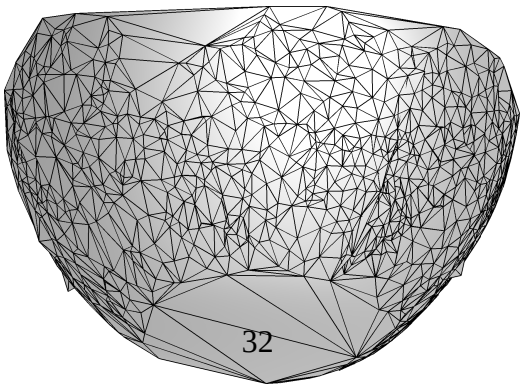
(a) Silicon phantom. Image taken from one of the cameras during a motion capture experiment.



(b) Camera geometry. The five cameras are positioned so that each point on the surface is viewed by two adjacent cameras.



(c) Points used in reconstruction. These are the images from the cameras, denoted 1 - 5 in order from the left. The points used from each pair to compute the 3D points are identified by colour. Note there is a gap in the coverage between cameras three and four. This is caused by a slight error in the calibration of one of the cameras causing matching points in the lower portion not to satisfy the epipolar constraint



(d) Reconstructed surface. The reconstructed points are triangulated for the purposes of visualisation

Figure 9: DIFT Breast Cancer screening case study

Camera pair	No. of correct matches	No. of false matches	Proportion of correct matches
$1 \leftrightarrow 2$	223	4	0.982
$2 \leftrightarrow 3$	252	1	0.996
$3 \leftrightarrow 4$	64	1	0.985
$4 \leftrightarrow 5$	166	4	0.977
$5 \leftrightarrow 1$	268	9	0.968

Table 2: Accuracy of estimated correspondences for each reconstruction pair in the laboratory case study

.

plementation in the vlFEAT open source library [18] was used to conduct a numerical experiment on one grayscale image pair. Of the top 223 correspondences computed, 120 were discarded manually because they included features not on the silicon phantom itself. Of the remaining 103 features detected, 12 were matched correctly, and the remaining features were entirely incorrect. Almost all of the detected SIFT keypoints used in the computed correspondences were located near the centre of one of the fiducial markers, but were matched with a marker in an entirely different location. The SIFT descriptors do not provide a good discriminator for features of the type used in this project, and hence perform very poorly for the correspondence problem. The successful matches were for cases where two or three markers were in very close proximity, or overlapping, creating a more distinctive shape on the surface.

5. Discussion

The results in the case study were produced with a MATLAB implementation of the algorithm, executed on a 2.53GHz dual-core laptop with 4GB RAM. The MATLAB version was R2011a in a 64-bit Linux environment. With this implementation, the full surface reconstruction procedure for the DIET system takes around 11 seconds, not including the extraction of the point features by threshold segmentation. There are numerous optimisations, not implemented here, that could be made to improve the efficiency of the code if required. 3D Nearest neighbour and range searches are frequently required. The efficiency of these can be increased significantly by storing the points in a binary space-partitioning data structure such as a kd-tree and using the $O(\log N)$ search algorithms that exist for these structures.

The two computational bottlenecks of the current implementation of the algorithm are the construction of the epipolar cloud, and the tangent plane fitting procedure. Both procedures exhibit embarrassing parallelism, and can easily be run in parallel on a multi-processor machine. It may also be possible to implement parts of both procedures on a GPU, however this is yet to be investigated.

Performance can also be improved by not attempting to construct tangent planes for every single point in the cloud. A number of start points can be chosen at random, and a search for connected components propagated from the neighbours of these points. With sufficiently many start points, the true surface component will be located with very high probability without having to estimate normals for every point in the epipolar cloud.

The algorithm, as presented, is only useful for smooth surfaces with a some prior knowledge about the surface geometry. It is conceivable that the algorithm

could be adapted to reconstruct a less restrictive class of surfaces. As discussed in Section 4.5, it requires little work to extend the algorithm to reconstruct piecewise smooth surfaces. An desirable extension to the algorithm would be to estimate the density ρ and curvature κ parameters adaptively based on local geometry, removing the need to specify these as prior information and allowing the reconstruction of surfaces whose geometry and density vary in different neighbourhoods.

Acknowledgements

Thanks to the Tertiary Education Commission (TEC) for providing funding through a Top Achiever Doctoral Scholarship for RGB. Thanks also to the anonymous referees for constructive reviews and for suggesting references and improvements to the content and presentation.

References

[1] A. Peters, S. Wortmann, R. Elliott, M. Staiger, J. Chase, E. V. Houten, Digital Image-based Elasto-Tomography: First experiments in surface based mechanical property estimation of gelatine phantoms, JSME International Journal 48 (2005) 562–569.

[2] A. Laurentini, The Visual Hull Concept for Silhouette-Based Image Understanding, IEEE Transactions on Pattern Analysis and Machine Intelligence 16 (2) (1994) 150–162.

[3] W. Matusik, C. Buehler, L. McMillan, Polyhedral Visual Hulls for Real-Time Rendering, Rendering Techniques 2001: Proceedings of the Eurographics Workshop in London, United Kingdom, June 25-27, 2001.

[4] M. Li, M. Magnor, H. Seidel, Hardware-Accelerated Visual Hull Reconstruction and Rendering, Graphics Interface Proceedings 2003: Canadian Human-Computer Communications Society.

[5] M. Tappen, W. Freeman, Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters, in: Ninth IEEE International Conference on Computer Vision, 2003. Proceedings, 2003, pp. 900–906.

[6] V. Kolmogorov, R. Zabih, Multi-camera scene reconstruction via graph cuts, Computer VisionECCV 2002 (2002) 8–40.

[7] J. Sun, H. Shum, N. Zheng, Stereo matching using belief propagation, Computer VisionECCV 2002 (2002) 450–452.

[8] D. Lowe, Distinctive image features from scale-invariant keypoints, International journal of computer vision 60 (2) (2004) 91–110.

[9] S. Smith, J. Brady, SUSANA new approach to low level image processing, International journal of computer vision 23 (1) (1997) 45–78.

[10] C. Harris, M. Stephens, A combined corner and edge detector, in: Alvey vision conference, Vol. 15, Manchester, UK, 1988, p. 50.

[11] C. Hernandez, F. Schmitt, Silhouette and stereo fusion for 3d object modelling, CVIU 96 (3) (2004) 367–392.

[12] J.-P. Pons, R. Keriven, O. Faugeras, Modelling dynamic scenes by registering multi-view image sequences, CVPR II (2005) 882–827.

[13] F. Dellaert, S. Seitz, C. Thorpe, S. Thrun, Structure from motion without correspondence, in: CVPR, Published by the IEEE Computer Society, 2000, p. 2557.

[14] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, Cambridge University Press, 2003.

[15] M. Fischler, R. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395.

[16] R. Brown, Three-dimensional motion capture for the DIET breast cancer imaging system, Ph.D. thesis, University of Canterbury (2008).

[17] T. Lotz, P. Simpson, D. Stocker, C. Hann, J. Chase, In vitro evaluation of surface based non-invasive breast cancer screening with Digital Image based Elasto Tomography (DIET), in: Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE, IEEE, 2010, pp. 3077–3080.

[18] A. Vedaldi, B. Fulkerson, VLFeat: An open and portable library of computer vision algorithms, `http://www.vlfeat.org/` (2008).